International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:7, 2007

Encoding and Compressing Data for Decreasing Number of Switches in Baseline Networks

Mohammad Ali Jabraeil Jamali, Ahmad Khademzadeh, Hasan Asil, and Amir Asil

Abstract**—**This method decrease usage power (expenditure) in networks on chips (NOC). This method data coding for data transferring in order to reduces expenditure. This method uses data compression reduces the size. Expenditure calculation in NOC occurs inside of NOC based on grown models and transitive activities in entry ports. The goal of simulating is to weigh expenditure for encoding, decoding and compressing in Baseline networks and reduction of switches in this type of networks.

Keywords**—**Networks on chip, Compression, Encoding, Baseline networks, Banyan networks.

I. Introduction

DEVELOPMENT of technology allows designers to use an Evolution system on chips. But complexity of such systems creates a difficult to inheritance and using their properties to growth and completing them. So designing of systems on chips (SOCS) which is based on using of their previous properties, by correlation of resources should manage together in a common confine, introduces some challenges for physical designing and way of changing system physical architecture.

A problem related to usage power in business is electrical capacity measured by long wires. But this problem has been solved by short wires and point to point at least in NOC and it's needed to spend expenditure. A method for reducing expenditure in NOC and each logical network is to reduce activities based on encoding and addressing. But such methods had suggested before 90th decade and they were used on bus architecture. We'll present a method by encoding and compressing for data transferring in Baseline Banyan networks. Moreover encoding by adding a compressor, it's tried to decrease size of data exchanged among switches. A point existed in designing of this system is not change of this chip based network. Evaluation of such system will be pointed in next parts. [3]

Mohammad Ali Jabraeil Jamali is with Islamic Azad University, Shabestar Branch, Iran (e-mail: m_jamali@itrc.ac.ir).

Ahmad khademzadeh is with Iran Telecommunication Research Center,Tehran, Iran (e-mail: zadeh@itrc.ac.ir).

Hasan Asil is with Islamic Azad University, Tabriz Branch (AzarShahr Branch), also with Islamic Azad University of Shabestar, Iran (e-mail:h.asil@asd.ir).

Amir Asil is with the Azarbaijan Software Developer, Iran (e-mail: a.asil@asd.ir).

II. Suggested Algorithm

The goal of this system is to reduce expenditure by encoding and also reduction in rate of data transferring by compressing and thereupon reduction in number of additional switches in network. In this research we've tried to reach this goal by merging networks on chips (NOC) and methods of data encoding and data compression. Coding data and decoding it in destination is the program. Notice that encoding and decoding are joint with compressing and we name them encoder and decoder.

As we know, data in network is transferred in packages and these packages are supported in routers by header (switching information) and a payload (data information). Now, two viewpoints are considered:

1. *Data encoding with header encoding*
2. *Data encoding without header encoding*

In first viewpoint size and length of encoded data are less than the second one. But this data requires expenditure for decoding in each switch. In second case data just should be decoded in main destination and decoding in path is not required. So we'll use the second method in our algorithm.

In next we'll present more details of suggested method.

III. Data Compressing Part

As we said, an encoding method and a compressing method are required. But which methods are suitable for encoding and compressing data? Figure of communication system is shown below:

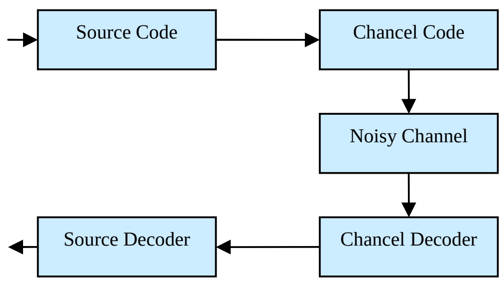

Fig. 1 Communication coding system pic [1]

In this system, Source encoder is used for compressing source data eliminating repetition of entered data. Also source decoder is used for decoding data and delivering it to destination.

In this system, channel encoder is used to create redundancy for data for error controlling and also channel decoder is used for opening channel encoder data. Because in this system the goal of data compressing is redundancy elimination in system or in other words the goal of sending coded data is sending them in shortest time, we've deleted channels in our system and just used source encoder and source decoder. So we observe suggested system Pic in Fig. 2:

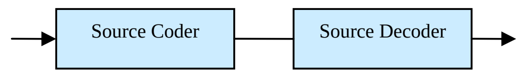

Fig. 2 Suggested encoding system pic

As it's specified in Fig. 2, we've used an encoder and a decoder for encoding and compressing data. But as we know, there are various methods available for compressing data. The research shows that there are more than 50 methods available for compressing data. Some of most famous methods are: Huffman, LZSS, LZW, Aritmatic etc. And some of applications like WinRAR & WinZIP use one of these methods. We tried to use features of Huffman method in this article. [13]

In computer sciences and theory of information, Huffman encoding is an encoding algorithm for wasteless data compression and this explanation refers to using tables with variable length code which are used for encoding each sign of resource. Variable length code table is reached by a special method based on each sign of resource occurrence possibility. This method has developed by David Huffman. He was PhD student of Philosophy in MIT University. He published an essay with the subject of "A method for producing code with the least redundancy" in 1952. Huffman coding uses a special method for the way of showing each sign. Huffman algorithm introduces a method for producing optimized prefix codes. Huffman code owns the shortest average length among other codes created on similar alphabet. In other words it's an optimized code.

This algorithm has introduced a method for producing optimized prefix codes and also owns shortest average length among other codes created on similar alphabet and this encoding is an optimized encoding. This is the reason made us to use this method. Available proves show that this encoding compresses data between 20 & 90 percent and whatever repetition gets more, output will get more, too.

In this method data encoding is performed based on number of signs redundancy and Huffman tree is created considering these signs. We use this method, too. But we use a table or collection of constant trees and encoding or decoding is performed on data. In these tree collections, type of encoding is realized proportionate with data and encoding or decoding is performed proportionate with the tree. The advantage of this method is reduction in expenditure of tree creating. The educed expenditure by this method equals N log N.

In this part we chose way of compressing data in algorithm and in next part we'll implement coding system on pointed system.

IV. Data Encoding Part

Development of technology allows designers to use an Evolution system on chips. But complexity of such systems creates a difficult to inheritance and using their properties to growth and completing them. So designing of systems on chip which is based on using of their previous properties, by correlation of resources should manage together in a common confine, introduces some challenges for physical designing and way of changing system physical architecture. A network on chips (NOC) consists of interior communication resources which have relation by channels. Rectifying noise errors in chip base networks and also reduction of data changing in these networks were considered. For decreasing them, encoding or decoding can be used in this algorithm. We've not presented an encoding method. We tried to implement LT algorithm presented by Han Vang from China in 2008 on this network. In fact we've placed encoding on leaf of Huffman's tree and after compressing, coding is performed automatically. Fig 3 shows main part of LT graph which uses XOR operator for coding. Expenditure in this algorithm is equal with increasing a bit of data. S_i shows entered data and C_i shows output of encoded data in system. [16]

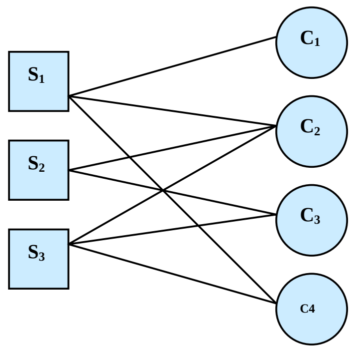

Fig. 3 Sample of LT graph [16]

V. Implementing Encoder and Decoder on Baseline Network

In this network we try to decrease usage power and size of data by changes we perform on switches. As we explained, encoding, decoding and compressing system in previous parts, in this part we implement them on this network. Way of adding encoder and decoder to switch has shown below.

Increasing requirement for communication services like data, voice, image and recent progresses in VLSI cause to generating BISDN networks.

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:7, 2007

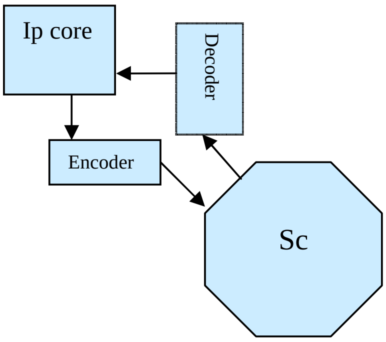

Fig. 4 Way of adding encoder and decoder to switch

Almost all implementations performed on all of decay resistant switches have redundancy and the expenditure of creating such switches is much and switching gets difficult because of redundancy and its complicated strategies. In Baseline switches each cell passes by Log N of middle stage and it owns on path to reach output. So it can be vulnerable against decaying. [4]

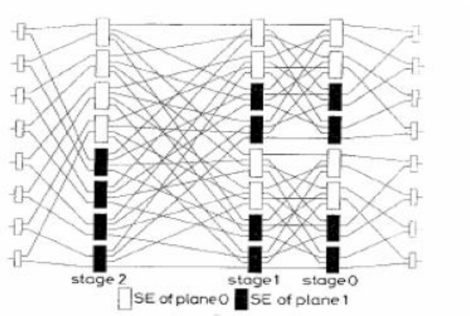

Fig. 5 Sample of Baseline network

As it's shown on Fig. 5, the most important problems in Banyan switches are reduction of expenditure and increasing resistance. As we mentioned, encoding and compressing data cause to decrease rate of data transferring and increasing resistance, but there was a point and it was: decreasing data in packages causes data perishing on transitive data. Now for elimination of empty packages and increasing resistance and reduction of expenditure in these resistant networks with deletion of additional switches and using two cores beside switches, we've tried to reduce expenditure spent on hardware. [5]

In next parts we're going to speak about simulating this network.

VI. System Evaluation

Some of encodings reduce expenditure only when the number of bosses are high or some of encodings have high level of efficiency when the number of data transferring are many, some of methods require knowledge of static parameters and interior traffic, but we use a method that it needs no one of above, in fact we use a common method. Base of this method is encodings performed on Bus networks. In these methods, method of encoding by decreasing average number of signal transferring has suggested strongly. In some of these methods some parameters of interior traffic is required, but in this research we simulate a method on Baseline networks which doesn't require such information.

Usually in encoding methods based on possibility, there's no need to know about network traffic, they act according to statistic flow. Intelligent bit is encoded according to it's past and real value and acts based on approximate statistic information.

In networks usually for calculating total time of a package delivery, the following formula is used:

$$TT = (ST + (NF - 1) * 2) * NP \quad (1)$$

TT: Total time without package incidence

ST: number of cycle-hour for performing judgeship and switching algorithm (10 cycles for Hermes NOC)

NF: number of flits (we should deficit the first flit because it's processed before)

NP: number of packages

For example for delivering 50 packages with 39 flits equals 4800 cycles per hour. [9]

Now we present this formula based on suggested algorithm. As we said before and considering available proves Huffman encoding usually reduces size of data between 20 & 90 percent and prevents repetition. But this system imposes an additional expenditure to network and this expenditure is time & usage power spent for encoding or decoding. So, the mentioned formula for this algorithm changes into:

$$TT = ((ST + (NC - 1) * 2) + Q) * NP \quad (2)$$

TT shows Total time without package incidence, **NC** shows coded system flits and its volume is less than **NF** between 20 & 90 percent.

Q is expenditure of data encoding as it's pointed in previous part in this system we've eliminated dynamic status of Huffman's tree, a static tree is spotted for encoding and we've implemented this encoding on whole network. By doing this the time spent to creating this tree (N*Log (N)) has been eliminated. Now if we place **NC** in above formula, **NC** will be like this:

$$NC <= (NC - NF / 5) \quad (3)$$

As it's clear, whatever the space between switches in network gets more, by multiplying length of path into this NC, the total time is reduced whereas the expenditure (Q) is constant.

Now we simulate this system and we show reached tentative results which are simulated totally object oriented.

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:7, 2007

Rate of data size reduction is showed below. Notice that the data is created randomly.

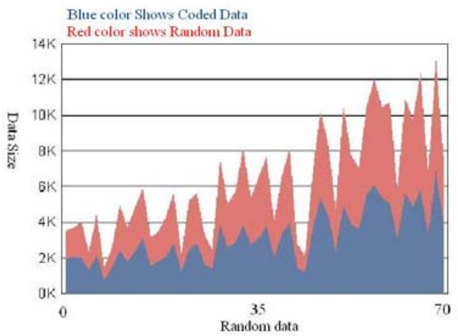

Fig. 6 Rate of reduction in size of random data

The most important points should spot in evaluation of this system are:

Adding an encoding part is an additional expenditure in encoding and decoding model, but this expenditure is recoverable by reducing traffic and data transferring rate.

Another important point is about rate of redundancy and similarity of codes to each other. Whatever redundancy of transitive data increases, the occupied volume is less than normal status. In this method of encoding usually rate of size reduction is between 20 & 60 percent (considering entered and transitive data). Also as it's shown on Fig. 4, whatever the space between resource and destination increases, system operative power increases by this encoding.

To make this method practical, we compared usage power in encoded system with the Baseline network shown on Fig. 5, results of simulating show that this method because of using LT encoding is more secure and uses less power than introduced switches in Fig. 5.

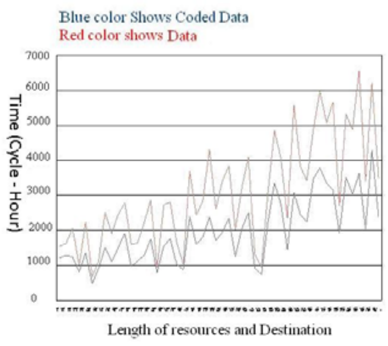

Fig. 7 Increasing efficiency by increasing path length of resources and destination of package

One of the other advantages of this switch is high level of security in transferring data. It transfers small size of data, increases security by encoding. Until now, we were speaking about high level of security in compressing and encoding. But as we know by reducing size of data on transitive data, a part of package is sent empty. For avoiding this case, we eliminated additional switches and (half of them) and we used two cores in one switch and we tried to make encoding and compressing system practical in chip based networks. By doing this number of switches decreased uttermost 50 percent. The following pic shows results of simulating this method on Banyan Baseline networks.

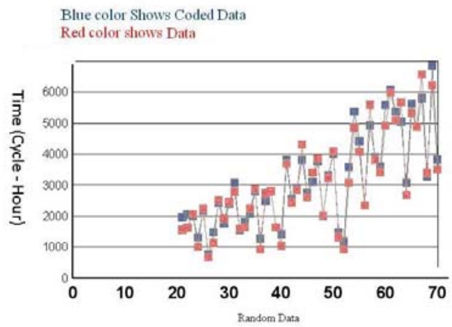

Fig. 8 Results of implementing suggested switch on Baseline networks and elimination of additional switches

As we see in Fig. 8, suggested chip based network almost have similar time and power in comparison to Banyan Baseline network whereas in suggested network, number of switches has reduced 50 percent.

VII. Result

In this article we changed method of data transferring in Banyan switches by data encoding and methods of compressing and by reducing data transferring among switches we eliminated additional switches. Notice that this type of switches are used mostly when data has high level of similarity and the usages are: image transferring etc. in this method moreover reducing number of switches by rate of 50 percent, we tried to increase confidence and security in data transferring. Also usage power was reduced when we had data transferring from one switch to another. This method has some deferment in encoding and decoding time. But by using other encoding methods or by Huffman's 2 adaptive encoding methods and Huffman's spread compressing we can increase data encoding and we can increase output by doing this in future.

References

[1] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das. A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In Proc. of ISCA, 2007.

[2] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. Hu, M. Irwin, M. Kandemir, and N. Vijaykrishnan. Leakage current: Moore's law meets static power. Computer, 36(12):65–77, December 2003.

[3] S. Kolson, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani. A NoC architecture and design methodology.In Proc. of ISVLSI, 2002.

[4] P. U. Tagle and N. K. Sharma, "Performance of Fault-Tolerant ATM Switch", IEE Proceedings on Communication, Vol. 143, No. 5 , pp. 318-324, October 1996
[5] Muhammad Anan and Mohsen Guizani “Modeling and Simulation of A Fault Tolerant ATM Switching Architecture,” proc. Of IEEE Simulation Symposium , pp. 42-47, 2000 .
[6] Neeraj K.Sharma “ Fault Tolerant Batcher Banyan Packet Switch” , proc. Of IEEE Computing and Communications Conference, New Jersey, IEEE 1997 pp: 125-130, 1997.
[7] W.Syen E.Chen,Y.M.Kim ,Y.W.Yao and M.T.Liu “FDB : A High Performance Fault Tolerant Switching Fabric for ATM Switching Systems, ” Proc. of the 10th IEEE International Phoenix Conference on Computers and communications, pp.703-709 , 1991 .
[8] soleymanpour m.javad “ Fault To levant Dedign of anyan Switch” ,Communications Conference, Ikt Jan, IEEE 2004.
[9] “Contribution of Packet Sizes to Packet and Byte Volumes,” MCI vBNS http://www.nlanr.net/NA/Learn/ packetsize.html.
[10] “Flow Counts as a Function of Flow Granularity,” FIX-West,http://www.nlanr.ner/NA/Learn/aggregation.html.
[11] N. McKeown, “Scheduling Algorithms for Input-Queued Cell Switches,”Ph.D. thesis, Univ. Calif. Berkeley, May 1995.
[12] "Huffman encoding Data" URL:http://www.byui.edu/ricks/employee_/C...5/chap10.5.pdf
[13] "Compressing Data" URL:http://ginger.cs.uwp.edu/C580/LZWovrhd.pdf
[14] N. McKeown, V. Anantharam, and J. Walrand, “Achieving 100% Throughput in an Input-Queued Switch,” Proc. IEEE INFOCOM ’96, San Francisco, CA, Mar. 1996.
[15] C. Labovitz, G. R. Malan, and F. Jahanian, “Internet Routing Instability,” Proc. ACM SIGCOMM ’97, Cannes, France, Sept. 1997.
[16] Han Wang, Hardware Designs for LT Coding, Faculty of Electrical Engineering, Mathematics and Computer Science, 2006